sponding to the component of the patch determined at step **1315** and the value corresponding to the component of the patch accompanying the MODIFY message, in accordance with Equation (1) described above.

[**0175**] If sequential modifications to the current component need to be multiplied, then execution continues at step **1435**. At step **1435**, the value of the current component is set to the weighted geometric mean of the component value corresponding to the component of the patch determined at step **1315** and the value corresponding to the component of the patch accompanying the MODIFY message, in accordance with Equation (2) described above.

[**0176**] If modifications to the current component are discrete (i.e. sequential modifications replace earlier component values by later values), then execution continues at **1440**. At step **1440**, the value of the current component is set is set to the result of a weighted election, as described above.

[**0177**] Following any one of steps **1430**, **1435** and **1440**, execution returns to step **1405** of the method **1400**.

[**0178**] A method **1500** of transmitting a LOGIN message, as executed at step **114** of the method **100**, will now be described with reference **FIG. 15**. The LOGIN message indicates that the sender of the message is logging into an editing session on the network **220**. The method **1500** is preferably implemented as software resident on the hard disk drive **210** and being controlled in its execution by the processor **205**.

[**0179**] The method **1500** begins at step **1105**, where the processor **205** constructs the LOGIN message. The LOGIN message includes the username of the collaborator who is logging into an editing session on the network **220** and the UUID of the document to which the collaborator is logging in. At the next step **1510**, the LOGIN message is transmitted to the active collaborators according to the list of active collaborators configured within memory **206**, in accordance with the method **1600** shown in **FIG. 16**.

[**0180**] A method **1600** of transmitting a message to the active collaborators listed in the document, as executed at steps **410**, **710**, **1110**, **1210**, **1510**, and **2010**, will now be described with reference to **FIG. 16**. The method **1600** is preferably implemented as software resident on the hard disk drive **210** and being controlled in its execution by the processor **205**.

[**0181**] The method **1600** begins at step **1605**, where if the processor **205** determines that there has not been an attempt to transmit the message to any one of the active collaborators according to the list of active collaborators configured within memory **206**, then the method **1600** proceeds to step **1610**. Otherwise execution returns to step **101** of the method **100**. At step **1610**, if the processor **205** determines that the collaborator identified at step **1605** is not online then the method **1600** returns to step **1605**. Otherwise, the method **1600** proceeds to step **1615**. At step **1615**, the message is transmitted over the network **220** to the collaborator identified at step **1605** and the method **1600** concludes.

[**0182**] A method **2000** of transmitting a LOGOUT message, as executed at step **116** of the method **100**, will now be described with reference **FIG. 20**. The LOGOUT message indicates that the sender of the message is logging out of an editing session for the document on the network **220**. The method **2000** is preferably implemented as software resident on the hard disk drive **210** and being controlled in its execution by the processor **205**.

[**0183**] The method **2000** begins at step **2005**, where the processor **205** constructs the LOGOUT message from a predetermined LOGOUT opcode. The LOGOUT message includes the username of the collaborator who is logging out of an editing session for the document on the network **220**, and the UUID of the document from which the user is logging out. At the next step **2010**, the LOGOUT message is transmitted to the collaborators who are currently online, in accordance with the method **1600** shown in **FIG. 16**.

[**0184**] The methods described above will now be described in further detail by way of example, with reference to **FIGS. 17, 18** and **19**.

[**0185**] **FIG. 17** is a sequence diagram showing the process of publication and acknowledgment of a document, followed by the logout of a collaborator. The collaborators are labelled X, Y and Z and are each operating a computer similar to the computer **200** and which are connected to the network **220**. In accordance with this example, X is operating the computer **200**, Y is operating the computer **250** and Z is operating the computer **252**. The arrows referenced as **1715**, **1720**, **1725**, **1730**, **1735**, and **1740** represent a publishing sequence, where X publishes a document and the collaborators Y and Z acknowledge receipt of the document.

[**0186**] Prior to publishing the document, the computer **200** being operated by X constructs the PUBLISH message from the predetermined PUBLISH message opcode and includes the username (X) of the publisher, a current version string (0), and the document to be published. At the line **1715**, the computer **200** transmits the publication message to the computer **250** being operated by Y. Then the computer **200** transmits the publication message to the computer **252** being operated by Z as represented by the arrow **1720**.

[**0187**] After receiving the PUBLISH message, the computer **250** being operated by Y constructs the acknowledgment message (i.e., ACK_DOC) from the predetermined ACK_DOC opcode, the username of the author of the document (i.e., X), the username (Y) of the acknowledger, a UUID contained in the received document, the current version string (0), and the result of hashing the document. Then the computer **250** being operated by Y transmits the ACK_DOC message to the computer **200**, as represented by the arrow **1725**. The computer **250** also transmits the ACK_DOC message to the computer **252** being operated by Z, as referenced by the arrow **1730**.

[**0188**] After receiving the ACK_DOC message the computer **252** constructs another acknowledgment message (i.e., ACK_DOC) from the predetermined ACK_DOC opcode, the username of the author of the document (i.e., X), the username (Z) of the acknowledger, the UUID contained in the received document, the current version string (0), and the result of hashing the document. Then the computer **252** transmits the ACK_DOC message to the computer **200**, as represented by the arrow **1735**. The computer **252** also transmits the ACK_DOC message to the computer **250** being operated by Y, as reference by the arrow **1740**.

[**0189**] As each of the collaborators X, Y and Z receive the ACK_DOC messages, the computers **200**, **250** and **252** being operated by the X, Y and Z, respectively, update their